Other Buds in Membrane Computing

Miguel A. Gutiérrez-Naranjo

Department of Computer Science and Artificial Intelligence, University of Seville,
Seville, Spain
magutier@us.es

Abstract. It is well-known the huge Mario's contribution to the development of Membrane Computing. Many researchers may relate his name to the theory of complexity classes in P systems, the research of frontiers of the tractability or the application of Membrane Computing to model real-life situations as the Quorum Sensing System in *Vibrio fischeri* or the Bearded Vulture ecosystem. Beyond these research areas, in the last years Mario has presented many new research lines which can be considered as *buds* in the robust Membrane Computing tree. Many of them were the origin of new research branches, but some others are still waiting to be developed. This paper revisits some of these *buds*.

1 Introduction

Mario has contributed to the development of Membrane Computing in many research lines (see, e.g., [16,24,57]). From his early works on the formalization of transition P systems [53] or the links between P systems and diophantine sets [61], Mario has published dozens of papers on Membrane Computing. His contributions cover many different areas of the P systems research, from theoretical ones to real-life case studies applications, together with the development of different simulators or the proposal of many different P systems models. Many young researchers may relate his name to the theory of complexity classes in P systems [51], the research of frontiers of the tractability [46] or to the application of Membrane Computing to model real-life situations as the Quorum Sensing System in *Vibrio fischeri* [59] or the Bearded Vulture ecosystem [9], but Mario's contributions go deeper in many other areas.

Beyond these strong branches in the robust Membrane Computing tree, there are many other research lines in Membrane Computing where Mario also has made a pioneer contribution. Some of these ideas were shortly developed in a few papers and others only appear in the paper where they were presented. I call *buds* these ideas. In this paper, I revisit some of these *buds* which are waiting for young researchers to be studied in depth.

The paper is organized as follows: In Sect. 2, some papers where Mario explored the links between Membrane Computing and Artificial Intelligence are revisited. They cover aspects related to sorting and searching algorithms and machine learning. Section 3 recalls other contributions bridging P systems with other aspects of Computer Science, as metrics defined on configurations of P systems, properties of Markov chains or the possibility of computing backwards. Later, Sect. 4 revisits some papers related to the graphical representation of P systems and Sect. 5 some other papers not included in the previous sections. Finally, some conclusions are drawn.

2 Artificial Intelligence

Mario is a full professor at the Department of Computer Science and Artificial Intelligence in the University of Seville, and both disciplines, Computer Science and Artificial Intelligence, have been present in Mario's research. We revisit some of these contributions.

2.1 Sorting

Sorting sequences of items according to specified criteria is on the basis on many computational processes and this is a recurrent problem in Membrane Computing (see, e.g. [1,2]). In the case of parallel algorithms, the problem is especially interesting, since it requires an appropriate combination of computation and communication.

In [14], two models for sorting sequences of numbers were presented. They were based on bitonic sorting networks. The key idea is the use of *bitonic merge-sort* which is a parallel algorithm for sorting introduced by Batcher [3]. The first of the models consists on N membranes, each of them storing two numbers; one number is an element of the sequence, and the other one is an auxiliary register used to route values. A number is encoded as the multiplicity of a symbol a in each membrane. Moreover, membranes are disposed on a $2D$-mesh, where only communication between neighboring membranes on the mesh is permitted. This model uses a variant of P systems called *P systems with dynamic communication graphs* [11], which is closely related to the implementation of the bitonic sort on the $2D$-mesh[1]. The second model presented in [14] consisted of only one membrane, where all the N numbers were encoded as occurrences of N different symbols. Restrictions on communication were no longer imposed, as if the underlying communication graph was the complete graph. Later, a new model which takes ideas from both models has been presented in [15].

2.2 Searching

The design of solutions for **NP**-complete problems in Membrane Computing usually trades time against space in order to solve these problems in polynomial (even linear) time with respect to the size of the input data [51]. The cost is the

[1] Such model of *P systems with dynamic communication graphs* is another of the many *buds* which is waiting to be explored. It follows the same lines as the one presented in [12,13].

number of resources, mainly the number of membranes, which grows exponentially. The usual idea of such brute force algorithms is to encode each feasible solution in one membrane. The number of candidate solutions is exponential in the input size, but the coding and checking process can be done in polynomial time. In spite of the theoretical success, such approaches are far from being applicable to real-world problems and other research lines must be explored. In this context, Mario has also made contributions by applying into the Membrane Computing framework some ideas from searching methods studied in Artificial Intelligence.

In [34], the problem of solving the N-queens puzzle with P systems was studied by considering a depth-first search strategy. Depth-first search is a well-known algorithm for exploring tree or graph data structures. One starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores as far as possible along each branch before backtracking. In the paper, the authors explore the possibilities of introducing such search strategy in the framework of Membrane Computing. The case study was the N-queens puzzle. This is an old well-known problem. It is a generalization of the 8-queens problem which consists of placing N chess queens on an $N \times N$ board. In [30], a first solution to the N-queens problem in Membrane Computing had been shown, but the solution was based on an appropriate encoding of the problem in a SAT formula and the use of a modified solution for the SAT problem with P systems. The same problem of N-queens also was considered in [35] where some ideas from local search were studied in the framework of Membrane Computing. In this paper, from an initial $N \times N$ chessboard with N queens, different movements of the queens are performed in order to *improve* the position. Such improvements are evaluated by using the notion of *collision* [63]: *The number of collisions on a diagonal line is one less than the number of queens on the line, if the line is not empty, and zero if the line is empty. The sum of collisions on all diagonal lines is the total number of collisions between queens.* The target of the computation is to move from a board to another, with the corresponding encoding, by decreasing the number of *collisions*. Figure 1, borrowed from [35], illustrates the process.

2.3 Machine Learning

Mario has also made contributions by linking Membrane Computing to Machine Learning. In this section, two of them are revisited. The first one takes ideas form *supervised learning* and the second one from *unsupervised learning*.

Spiking neural P systems (SN P systems) were introduced in 2006 (see [42]) with the aim of incorporating in Membrane Computing ideas specific to spike-based neural networks. Only two years later, a first model for Hebbian learning with spiking neural P systems was presented [32]. The target of this paper was to explore the applicability of ideas from the artificial neural networks into the SN P systems. Artificial neural networks [40] is one of the most powerful tools in Machine Learning and the most extended learning algorithm for such networks, backpropagation [62], can be roughly described as the iterative refinement of the weights associated to the synapses among neurons in order to minimize a

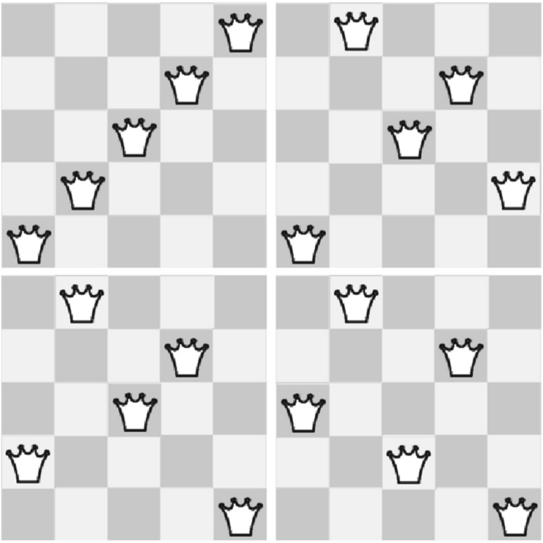

Fig. 1. Starting from a configuration C_0 with 4 collisions (up-left) we can reach C_1 with 3 collisions (up-right) and then C_2 with 2 collisions (bottom-left) and finally C_3 with 0 collisions (bottom-right), which is a solution to the 5-queens problem. Figure borrowed from [35].

loss function. Such change in the weights is inspired in Donald Hebb's works [41] and all the learning processes based on these principles are called *Hebbian Learning*. In [32], a first approach to Hebbian learning in the framework of Membrane Computing was presented. In this paper, a new feature coming from alive neurons was added to the SN P systems: the *decay*. Such decay of the electric potential inside an alive neuron along time was codified by endowing the rules with a finite non-increasing sequence of natural numbers called the decaying sequence. Besides these sequences, the learning model is structured in *Hebbian SN P system units*, which consider weights associated to the synapses between neurons. According to the learning process inspired by Hebb's work, the weights change along time according to the concept of *efficacy* introduced in the paper. In this way, the efficacy of the synapses with a high contribution to reach the spiking threshold is increased and, on the other hand, if the synapses has no contribution (or it is low) the efficacy is decreased.

The second *bud* revisited in this section is related to the data clustering problem. In the paper [32] discussed above, Mario made an exploration by bridging Membrane Computing with supervised learning. The proposal in [50] links P systems with unsupervised learning. The clustering algorithm presented in this paper is based on a tissue-like P system with a loop structure of cells. The objects of the cells express the candidate cluster centers and are evolved by the evolution rules. Based on the loop membrane structure, the communication rules realize a local neighborhood topology, which helps the co-evolution of the objects and improves the diversity of objects in the system. The tissue-like P system can effectively search for the optimal clustering partition with the help of its parallel computing advantage. The proposed clustering algorithm is evaluated on four artificial data sets and six real-life data sets. Experimental results show that the proposed clustering algorithm is superior or competitive to classical k-means algorithm and other evolutionary clustering algorithms.

3 Other Areas of Computer Science

Besides Artificial Intelligence, Mario has also contributed to bridge Membrane Computing with many other areas in Computer Science.

As a first example, we can consider the study of metrics on configurations presented in [19]. In this paper, two different (weak) metrics were presented. The first one was based on the distance between regions. The distance between two regions was defined as the cardinality of the symmetrical difference of their associated multisets. This definition was used to measure the distance between two occurrences of the same membrane in two different configurations and the difference between configurations is the sum of the differences between their regions. For the definition of the second weak metric, a new auxiliary concept called *dependency graph* was defined. This concept has been widely used for studying frontiers on complexity classes (see, e.g., [38]), but it was firstly defined in this paper for studying the proximity between configurations. The distance between two nodes of the dependency graph is the basis of the study of proximity between configurations.

Other remarkable Mario's contribution which is still a *bud* in the Membrane Computing tree is related to the link between P systems and Markov chains. In [10], the authors propose a first approach to the problem of computing the natural powers of the transition matrix of a finite and homogeneous Markov chain. Such computation allows to estimate its limit in the case that it is convergent, and therefore, to know the stationary distribution of the process. This subject has been treated with other bio-inspired techniques in [7] where two algorithms based on DNA were described. The proposed cellular computing solution provides an exact solution in a time which is linear in the order of the power and it is independent of the number of states of the Markov chain. Such Markov chains were also considered later in [8]. In this paper, the aperiodicity of irreducible Markov chains was characterized by using P systems.

A different question was studied in [33], where the problem of computing backwards with P systems was considered. The starting point for this study was to look for the previous state of a given one in a computational model where the time is considered in a discrete way. In this study, the authors consider a variant of the dependency graphs [18] for introducing a representation of the computation of a P system based on matrices. This matrix representation opens a door for the study of algebraic properties of cell-like and tissue-like P systems and represents one of the most promising *buds* waiting to be explored. Mario has also contributed to the study of the matrix representation of spiking neural P systems [66] which is the basis of the efficient simulation of such devices [5].

4 Graphics

Many P system models allow to change their membrane structure by adding new membranes (e.g., by the application of *creation* or *division* rules [52]) or removing membranes (by the application of *dissolution* rules [39]). This evolution in time of the membrane structure of a P system is the starting point for studying the evolution of graphical structures.

The first approach for linking the computation of P systems with the evolution of graphical structures was presented in [25,26]. In these papers, P systems were used to simulate the growth and development of living plants. This approach mixes L systems [23,45] and P systems. A different approach was presented in [60]. In this paper, the growth of branching structures was studied by using exclusively P systems. The key idea is the use of a cell-like P system model with evolution and creation rules. The membrane structure of a cell-like model is a tree-like graph which is easily visualized as a branching structure. The geometrical properties of the associated picture can be obtained by the association of a *meaning* to the objects in the multisets in the different membranes. For example, each membrane represents a segment in the corresponding picture and the length of the drawn segment depends on the multiplicity of a *length-unit* object. These ideas were also considered in [58], where a specific software for this graphical representation was developed. Some examples of polygons, spirals, friezes and plants can be found in this paper. Figure 2 shows some of them.

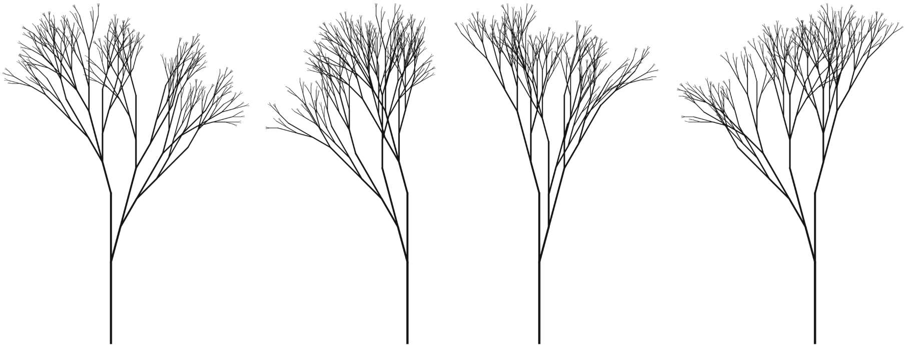

Fig. 2. Graphical representation of four configurations of a P system. Images borrowed from [58].

A different approach was presented in [31]. It is also related to the graphical representation of P systems, but it explores the possibility of Membrane Computing devices for representing fractals. The starting point here is that a fractal can be considered, roughly speaking, as a self-similar geometrical structure which can be generated by the application of (infinitely repeated) fixed rules. In this way, the generation of fractals can be associated, on the one hand, to the evolution of P systems by using creation rules for obtaining a new (and more precise) stage in the generation of the fractal and, on the other hand, to the *interpretation* of the symbols inside the membranes for representing the *geometrical information* of the fractal. The paper presents a pair of classic fractals, the Koch curve [43,44] and the middle third Cantor set [6], in the framework of P systems. Besides this mathematical objects, the paper also points out the possibility of using the non-determinism of P systems for studying random fractals, which can be considered as the formal representation of many real-life objects with fractal dimension[2].

5 Other Buds

The contributions made by Mario cover many different areas, some of them by bridging Membrane Computing with other unexpected research fields. One of these *rara avis* can be found in [28] where a Membrane Computing model for ballistic depositions was presented. The starting point in this study is the evolution of *rough interfaces* between different media. The propagation of forest fires [17] or the growth of a colony of bacteria [4] are examples showing such interfaces, although all surfaces in Nature can be seen as rough surfaces, since the concept of roughness is associated to the scale and all the natural surfaces are rough at an appropriate scale. The evolution of a surface can be modelled by the concepts of *erosion*, where some elements are removed, or *deposition*, where new elements are placed. Ballistic Deposition was proposed by Vold [65] and Sutherland [64] as a model where the particle follows a straight vertical trajectory until it reaches the surface, whereupon it sticks (see Fig. 3). In [28] the problem was modelled by a tissue-like P system model with a linear membrane structure where each cell represents a column of the aggregate and the pieces of information needed for encoding the growth process are encoded by means of the multisets of objects in the cells.

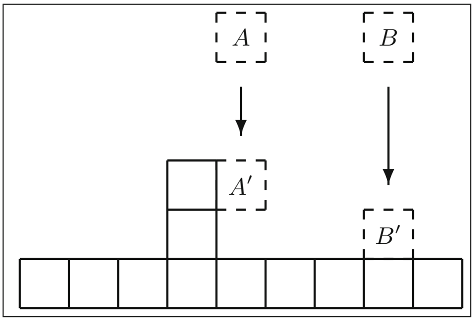

Fig. 3. Ballistic Deposition. Figure borrowed from [28].

[2] These ideas were also considered in [37], but it is still one of the most promising *buds* waiting to be developed.

A further exploration looking for links between Membrane Computing and other research areas was presented in [54]. In this case, the target was to bridge P systems and *reaction systems*. Reaction systems, also known as *R systems*, is a bio-inspired computation model [21,22] which shares with P systems some features as the use of populations of reactants (molecules) which evolve by means of reactions. This paper compares the two computation models, and further results can be found in [56].

Another theoretical *bud* can be found in [36]. In this paper, the degree of parallelism in P systems is investigated. The starting point is to study different tools for comparing the designs of P systems able to perform the same task. Two designs can be compared according to many different criteria. In this paper, the authors focus the attention on the parallelism. In this way, a *bad* design of a P system consists of a P system which does not exploit its parallelism, that is, it works as a sequential machine: in each step only one object evolve in one membrane whereas the remaining objects do not evolve. On the other hand, a *good* design consists of a P system in which a huge amount of objects are evolving simultaneously in all membranes. If both P systems perform the same task, it is obvious that the second one is a better design than the first one.

6 Conclusions

Mario's contribution to the development of Membrane Computing has been enormous and in the last years he has been one of the pillars of the Membrane Computing community[3]. The research lines opened by him cover all the research fields in Membrane Computing. Many of his proposals have been studied in depth, but some of them are still waiting for research efforts. In this paper, we have revisited some of them but many others have not been cited in the previous sections. I encourage the young researchers to read Mario's contributions cited in this paper, but not only them. Many other papers not cited here contain seminal ideas which are waiting for young and enthusiastic researchers who can help these *buds* develop and become new strong branches in the Membrane Computing tree.

References

1. Alhazov, A., Sburlan, D.: Static sorting P systems. In: Ciobanu et al. [16], pp. 215–252. https://doi.org/10.1007/3-540-29937-8_8
2. Ardelean, I.I., Ceterchi, R., Tomescu, A.I.: Sorting with P systems: a biological perspective. Rom. J. Inf. Sci. Technol. 11(3), 243–252 (2008). http://www.imt.ro/romjist/Volum11/Number11_3/pdf/03-Ardelean.pdf

[3] At the moment of writing this paper (April 2018), according to Scopus, Mario has published in collaboration with 150 co-authors. This fact can give an idea of Mario's role as an *engine* of the P systems community.

3. Batcher, K.E.: Sorting networks and their applications. In: American Federation of Information Processing Societies: AFIPS Conference Proceedings: 1968 Spring Joint Computer Conference, Atlantic City, NJ, USA, vol. 32, pp. 307–314. Thomson Book Company, Washington D.C. http://doi.acm.org/10.1145/1468075.1468121
4. Ben-Jacob, E., Schochet, O., Tenenbaum, A., Cohen, I., Czirok, A., Vicsek, T.: Generic modelling of cooperative growth patterns in bacterial colonies. Nature **368**(6466), 46–49 (1994). https://www.nature.com/articles/368046a0
5. Cabarle, F.G.C., Adorna, H., Martínez, M.A.: A spiking neural P system simulator based on CUDA. In: Gheorghe, M., Păun, G., Rozenberg, G., Salomaa, A., Verlan, S. (eds.) CMC 2011. LNCS, vol. 7184, pp. 87–103. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28024-5_8
6. Cantor, G.: Über unendliche, lineare punktmannichfaltigkeiten v. Math. Ann. **21**, 545–591 (1883). https://doi.org/10.1007/BF01446819
7. Cardona, M., Colomer, M., Conde, J., Miret, J., Miró, J., Zaragoza, A.: Markov chains: computing limit existence and approximations with DNA. Biosystems **81**(3), 261–266 (2005). http://www.sciencedirect.com/science/article/pii/S0303264705000663
8. Cardona, M., Colomer, M.A., Pérez-Jiménez, M.J.: Characterizing the aperiodicity of irreducible Markov chains by using P systems. In: Gutiérrez-Escudero et al. [29], pp. 81–95, https://idus.us.es/xmlui/handle/11441/38857
9. Cardona, M., Colomer, M.A., Pérez-Jiménez, M.J., Sanuy, D., Margalida, A.: Modeling ecosystems using P systems: the bearded vulture, a case study. In: Corne, D.W., Frisco, P., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2008. LNCS, vol. 5391, pp. 137–156. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-95885-7_11
10. Cardona, M., Colomer, M.Á., Pérez-Jiménez, M.J., Zaragoza, A.: Handling Markov chains with membrane computing. In: Gutiérrez-Naranjo, M.A., et al. (eds.) Fourth Brainstorming Week on Membrane Computing, vol. I, pp. 99–111. Fénix Editora, Sevilla (2006). https://idus.us.es/xmlui/handle/11441/36996
11. Ceterchi, R., Pérez-Jiménez, M.J.: On two-dimensional mesh networks and their simulation with P systems. In: Mauri et al. [49], pp. 259–277. https://doi.org/10.1007/978-3-540-31837-8_15
12. Ceterchi, R., Pérez-Jiménez, M.J.: A perfect shuffle algorithm for reduction processes and its simulation with P systems. In: International Conference on Computers and Communications, pp. 92–97 (2004). http://www.cs.us.es/~marper/investigacion/oradea.pdf
13. Ceterchi, R., Pérez-Jiménez, M.J.: Simulating shuffle-exchange networks with P systems. In: Păun et al. [55], pp. 117–129. https://idus.us.es/xmlui/handle/11441/34641
14. Ceterchi, R., Pérez-Jiménez, M.J., Tomescu, A.I.: Simulating the bitonic sort using P systems. In: Eleftherakis, G., Kefalas, P., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2007. LNCS, vol. 4860, pp. 172–192. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77312-2_11
15. Ceterchi, R., Pérez-Jiménez, M.J., Tomescu, A.I.: Sorting omega networks simulated with P systems: optimal data layouts. In: Díaz-Pernil et al. [20], pp. 79–92. https://idus.us.es/xmlui/bitstream/handle/11441/68018/dl-ps-4a.pdf
16. Ciobanu, G., Pérez-Jiménez, M.J., Păun, Gh. (eds.): Applications of Membrane Computing. Natural Computing. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-29937-8

17. Clar, S., Drossel, B., Schwabl, F.: Forest fires and other examples of self-organized criticality. J. Phys.: Condens. Matter **8**(37), 6803 (1996). http://stacks.iop.org/0953-8984/8/i=37/a=004
18. Cordón-Franco, A., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A.: Exploring computation trees associated with P systems. In: Mauri et al. [49], pp. 278–286. https://doi.org/10.1007/978-3-540-31837-8_16
19. Cordón-Franco, A., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A.: Weak metrics on configurations of a P system. In: Păun et al. [55], pp. 139–151. https://idus.us.es/xmlui/handle/11441/34641
20. Díaz-Pernil, D., Graciani, C., Gutiérrez-Naranjo, M.A., Păun, Gh., Pérez-Hurtado, I., Riscos-Núñez, A. (eds.): Sixth Brainstorming Week on Membrane Computing. Fénix Editora, Sevilla (2008)
21. Ehrenfeucht, A., Rozenberg, G.: Basic notions of reaction systems. In: Calude, C.S., Calude, E., Dinneen, M.J. (eds.) DLT 2004. LNCS, vol. 3340, pp. 27–29. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30550-7_3
22. Ehrenfeucht, A., Rozenberg, G.: Reaction systems. Fundam. Inform. **75**(1–4), 263–280 (2007). http://content.iospress.com/articles/fundamenta-informaticae/fi75-1-4-15
23. Frijters, D., Lindenmayer, A.: A model for the growth and flowering of aster novae-angliae on the basis of table <1, 0> L-systems. In: Rozenberg, G., Salomaa, A. (eds.) L Systems. LNCS, vol. 15, pp. 24–52. Springer, Heidelberg (1974). https://doi.org/10.1007/3-540-06867-8_2
24. Frisco, P., Gheorghe, M., Pérez-Jiménez, M.J.: Applications of Membrane Computing in Systems and Synthetic Biology. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-03191-0
25. Georgiou, A., Gheorghe, M.: Generative devices used in graphics. In: Alhazov, A., Martín-Vide, C., Păun, Gh. (eds.) Preproceedings of the Workshop on Membrane Computing. Technical report, vol. 28/03, pp. 266–272. Research Group on Mathematical Linguistics, Universitat Rovira i Virgili, Tarragona (2003)
26. Georgiou, A., Gheorghe, M., Bernardini, F.: Membrane-based devices used in computer graphics. In: Ciobanu, G., Păun, Gh., Pérez-Jiménez, M.J. (eds.) Applications of Membrane Computing. Natural Computing, pp. 253–281. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-29937-8_9
27. Graciani, C., Păun, Gh., Romero-Jiménez, A., Sancho-Caparrini, F. (eds.): Fourth Brainstorming Week on Membrane Computing, vol. II. Fénix Editora, Sevilla (2006)
28. Graciani-Díaz, C., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: A membrane computing model for ballistic depositions. In: Gutiérrez-Naranjo, M.A. et al. (eds.) Fifth Brainstorming Week on Membrane Computing, pp. 179–197. Fénix Editora, Sevilla (2007). https://idus.us.es/xmlui/handle/11441/38583
29. Gutiérrez-Escudero, R., Gutiérrez-Naranjo, M.A., Păun, Gh., Pérez-Hurtado, I., Riscos-Núñez, A. (eds.): Seventh Brainstorming Week on Membrane Computing, vol. I. Fénix Editora, Sevilla (2009)
30. Gutiérrez-Naranjo, M.A., Martínez-del-Amor, M.A., Pérez-Hurtado, I., Pérez-Jiménez, M.J.: Solving the N-queens puzzle with P systems. In: Gutiérrez-Escudero et al. [29], pp. 199–210. https://idus.us.es/xmlui/handle/11441/38865
31. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Fractals and P systems. In: Graciani et al. [27], pp. 65–86. https://idus.us.es/xmlui/handle/11441/38328
32. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: A first model for Hebbian learning with spiking neural P systems. In: Díaz-Pernil et al. [20], pp. 211–233. https://idus.us.es/xmlui/handle/11441/38778

33. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Computing backwards with P systems. In: Gutiérrez-Escudero et al. [29], pp. 211–226. https://idus.us.es/xmlui/handle/11441/38866
34. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Membrane computing meets artificial intelligence: a case study. In: Martínez del Amor et al. [47], pp. 133–144. https://idus.us.es/xmlui/handle/11441/39004
35. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Implementing local search with membrane computing. In: Martínez del Amor et al. [48], pp. 159–168. https://idus.us.es/xmlui/handle/11441/39476
36. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A.: An approach to the degree of parallelism in P systems. In: Graciani-Díaz et al. [27], pp. 87–104. https://idus.us.es/xmlui/handle/11441/38335
37. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A., Romero-Campero, F.J.: How to express tumours using membrane systems. Prog. Nat. Sci. **17**(4), 449–457 (2007). http://www.tandfonline.com/doi/abs/10.1080/10020070708541022
38. Gutiérrez–Naranjo, M.A., Pérez–Jiménez, M.J., Riscos–Núñez, A., Romero–Campero, F.J.: On the power of dissolution in P systems with active membranes. In: Freund, R., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2005. LNCS, vol. 3850, pp. 224–240. Springer, Heidelberg (2006). https://doi.org/10.1007/11603047_16
39. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A., Romero-Campero, F.J.: Computational efficiency of dissolution rules in membrane systems. Int. J. Comput. Math. **83**(7), 593–611 (2006). https://doi.org/10.1080/00207160601065413
40. Haykin, S.S.: Neural Networks and Learning Machines, 3rd edn. Pearson Education, Upper Saddle River (2009)
41. Hebb, D.O.: The Organization of Behavior: A Neuropsychological Theory. Wiley, New York (1949)
42. Ionescu, M., Păun, Gh., Yokomori, T.: Spiking neural P systems. Fundam. Inform. **71**(2–3), 279–308 (2006). http://content.iospress.com/articles/fundamenta-informaticae/fi71-2-3-08
43. Koch, H.v.: Sur une courbe continue sans tangente, obtenue par une construction géométrique élémentaire. Arkiv för Matematik, Astronomi och Fysik **1**, 681–704 (1904)
44. von Koch, H.: Une méthode géometrique élémentaire pour l'étude de certaines questions de la théorie des courbes planes. Acta Math. **30**, 145–174 (1906). https://doi.org/10.1007/BF02418570
45. Lindenmayer, A.: Developmental systems without cellular interactions, their languages and grammars. J. Theor. Biol. **30**(3), 455–484 (1971)
46. Macías-Ramos, L.F., Martínez-del-Amor, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A., Valencia-Cabrera, L.: Unconventional approaches to tackle the P Versus NP problem. In: Păun, Gh., Rozenberg, G., Salomaa, A. (eds.) Discrete Mathematics and Computer Science. Memoriam Alexandru Mateescu (1952–2005), pp. 223–237. The Publishing House of the Romanian Academy (2014)
47. Martínez del Amor, M.A., Păun, Gh., Pérez Hurtado, I., Riscos-Núñez, A. (eds.): Eighth Brainstorming Week on Membrane Computing. Fénix Editora, Sevilla (2010)
48. Martínez del Amor, M.A., Păun, Gh., Pérez Hurtado, I., Romero Campero, F.J., Valencia-Cabrera, L. (eds.): Ninth Brainstorming Week on Membrane Computing. Fénix Editora, Sevilla (2011)

49. Mauri, G., Păun, Gh., Pérez-Jiménez, M.J., Rozenberg, G., Salomaa, A. (eds.): WMC 2004. LNCS, vol. 3365. Springer, Heidelberg (2005). https://doi.org/10.1007/b106721
50. Peng, H., Zhang, J., Wang, J., Wang, T., Pérez-Jiménez, M.J., Riscos-Núñez, A.: Membrane clustering: a novel clustering algorithm under membrane computing. In: Macías-Ramos, L.F. et al. (eds.) Twelfth Brainstorming Week on Membrane Computing, pp. 311–327. Fénix Editora, Sevilla (2014). https://idus.us.es/xmlui/handle/11441/33577
51. Pérez–Jiménez, M.J.: A computational complexity theory in membrane computing. In: Păun, Gh., Pérez-Jiménez, M.J., Riscos-Núñez, A., Rozenberg, G., Salomaa, A. (eds.) WMC 2009. LNCS, vol. 5957, pp. 125–148. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11467-0_10
52. Pérez-Jiménez, M.J., Riscos-Núñez, A., Romero-Jiménez, A., Woods, D.: Complexity - membrane division, membrane creation. In: Păun et al. [57], pp. 302–336
53. Pérez-Jiménez, M.J., Sancho-Caparrini, F.: A formalization of transition P systems. Fundam. Inform. **49**(1–3), 261–272 (2002). http://content.iospress.com/articles/fundamenta-informaticae/fi49-1-3-20
54. Păun, Gh., Pérez-Jiménez, M.J.: Towards bridging two cell-inspired models: P systems and R systems. In: Martínez del Amor et al. [48], pp. 305–316. https://idus.us.es/xmlui/handle/11441/39575
55. Păun, Gh., Riscos-Núñez, A., Romero-Jiménez, Á., Sancho-Caparrini, F. (eds.): Second Brainstorming Week on Membrane Computing. Fénix Editora, Sevilla (2004)
56. Păun, Gh., Pérez-Jiménez, M.J., Rozenberg, G.: Bridging membrane and reaction systems – Further results and research topics. In: Valencia-Cabrera, L., et al. (eds.) Eleventh Brainstorming Week on Membrane Computing, pp. 243–256. Fénix Editora, Sevilla, February 2013. https://idus.us.es/xmlui/handle/11441/33819
57. Păun, Gh., Rozenberg, G., Salomaa, A. (eds.): The Oxford Handbook of Membrane Computing. Oxford University Press, Oxford (2010)
58. Rivero-Gil, E., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Graphics and P systems: experiments with JPLANT. In: Díaz-Pernil et al. [20], pp. 241–253. https://idus.us.es/xmlui/handle/11441/38787
59. Romero-Campero, F.J., Pérez-Jiménez, M.J.: A model of the quorum sensing system in Vibrio fischeri using P systems. Artif. Life **14**(1), 95–109 (2008). https://doi.org/10.1162/artl.2008.14.1.95
60. Romero-Jiménez, A., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: The growth of branching structures with P systems. In: Graciani et al. [27], pp. 253–265. http://www.gcn.us.es/4BWMC/vol2/thegrowth.pdf
61. Jiménez, Á.R., Pérez Jiménez, M.J.: Generation of diophantine sets by computing P systems with external output. In: Calude, C.S., Dinneen, M.J., Peper, F. (eds.) UMC 2002. LNCS, vol. 2509, pp. 176–190. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45833-6_15
62. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**, 533–536 (1986)
63. Sosic, R., Gu, J.: Efficient local search with conflict minimization: a case study of the N-queens problem. IEEE Trans. Knowl. Data Eng. **6**(5), 661–668 (1994)
64. Sutherland, D.: Comments on Vold's simulation of floc formation. J. Colloid Interface Sci. **22**(3), 300–302 (1966). http://www.sciencedirect.com/science/article/pii/0021979766900373

65. Vold, M.J.: A numerical approach to the problem of sediment volume. J. Colloid Sci. **14**(2), 168–174 (1959). http://www.sciencedirect.com/science/article/pii/0095852259900418
66. Zeng, X., Adorna, H., Martínez-del-Amor, M.Á., Pan, L., Pérez-Jiménez, M.J.: Matrix representation of spiking neural P systems. In: Gheorghe, M., Hinze, T., Păun, G., Rozenberg, G., Salomaa, A. (eds.) CMC 2010. LNCS, vol. 6501, pp. 377–391. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-18123-8_29